The Structure and Complexity of Credal Semantics

Fabio G. Cozman, Denis D. Mauá
Universidade de São Paulo

September 13, 2016

Overview

1. Credal semantics: definition.
2. The structure of credal semantics.
3. Inference and query complexity.

Probabilistic logic programs

- A probabilistic logic program is a pair $\langle \mathbf{P}, \mathbf{PF} \rangle$ where
  - **P** is a normal logic program (no functions) and
  - **PF** is a set of probabilistic facts.

- Predicate r, atom $\mathrm{r}(\mathrm{t}_1, \ldots, \mathrm{t}_k)$, rule

$$A_0 :- A_1, \ldots, A_m, \mathbf{not}\ A_{m+1}, \ldots, \mathbf{not}\ A_n.$$

- A_0 is the head, the right hand side is the body.
- A rule without a body is a *fact*.
- A program without **not** is *definite*.
- Atom without logical variables is a *ground atom*.
- A program without logical variables is *propositional*.

Probabilistic facts

- A probabilistic fact is a fact associated with a probability:

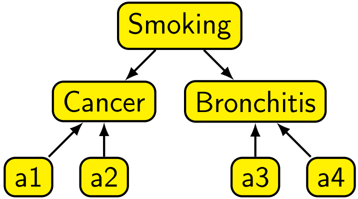
$$\mathbb{P}(A) = \alpha.$$

- Probabilistic facts are assumed independent.

Example: the Bayesian network Asia

- Predicates smoking, cancer, and bronchitis.
- Probabilistic logic program (ProbLog notation):

$$
\begin{gathered}
0.5 :: \text{smoking}. \\
\text{cancer} :- \text{smoking}, \text{a1}. \\
\text{cancer} :- \textbf{not}\ \text{smoking}, \text{a2}. \\
\text{bronchitis} :- \text{smoking}, \text{a3}. \\
\text{bronchitis} :- \textbf{not}\ \text{smoking}, \text{a4}. \\
0.1 :: \text{a1}. \qquad 0.01 :: \text{a2}. \\
0.6 :: \text{a3}. \qquad 0.3 :: \text{a4}.
\end{gathered}
$$



Acyclic programs (acyclic dependency graphs)

- For each total choice of probabilistic facts, we have an acyclic logic program (with the usual semantics).
- Hence the semantics of an acyclic probabilistic logic program is a single distribution — a Bayesian network (Poole (1993)).

Stratified programs:

- ... the grounded dependency graph has no cycle containing a *negative* edge.
- Example:

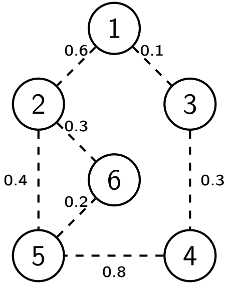
$$\begin{gathered}\text{path}(X,Y) :- \text{edge}(X,Y).\\ \text{path}(X,Y) :- \text{edge}(X,Y), \text{path}(X,Y).\end{gathered}$$

$$\begin{gathered}0.6 :: \text{edge}(1,2). \quad 0.1 :: \text{edge}(1,3).\\ 0.4 :: \text{edge}(2,5). \quad 0.3 :: \text{edge}(2,6).\\ 0.3 :: \text{edge}(3,4). \quad 0.8 :: \text{edge}(4,5).\\ 0.2 :: \text{edge}(5,6).\end{gathered}$$

A random graph

$$\text{path}(X, Y) :\!- \text{edge}(X, Y).$$
$$\text{path}(X, Y) :\!- \text{edge}(X, Y), \text{path}(X, Y).$$

$$0.6 :: \text{edge}(1, 2). \quad 0.1 :: \text{edge}(1, 3).$$
$$0.4 :: \text{edge}(2, 5). \quad 0.3 :: \text{edge}(2, 6).$$
$$0.3 :: \text{edge}(3, 4). \quad 0.8 :: \text{edge}(4, 5).$$
$$0.2 :: \text{edge}(5, 6).$$



Semantics of stratified probabilistic logic programs

- For each total choice of probabilistic facts, we have a stratified logic program.
- Hence the semantics of a stratified probabilistic logic program is a single distribution (with the usual semantics).

A word on semantics

- The semantics of acyclic and stratified programs is uncontroversial: just take the unique stable model (= answer set = well-founded model).

- To recap:
  - Consider logic program **P**.
  - For some interpretation $\mathcal{I}$, take the reduct $\mathbf{P}^{\mathcal{I}}$:
    - Ground **P**.
    - Remove rules with subgoal **not** A and $A \in \mathcal{I}$.
    - Remove subgoals **not** A from remaining rules.
  - Interpretation $\mathcal{I}$ is stable model if $\mathcal{I}$ is the minimal model of $\mathbf{P}^{\mathcal{I}}$.

Non-stratified program (cycle with negative edge)

- Non-stratified program may have more than one stable model.

The Dilbert example

$$\text{single}(X) :- \text{man}(X), \textbf{not } \text{husband}(X).$$

$$\text{husband}(X) :- \text{man}(X), \textbf{not } \text{single}(X).$$

$$0.9 :: \text{man}(\text{dilbert}).$$

- man(dilbert) is false: a unique stable model s_1.
- man(dilbert) is true: there are two stable models,

$$s_2 = \{\text{husband}(\text{dilbert}) = \text{true}, \text{single}(\text{dilbert}) = \text{false}\},$$

and

$$s_3 = \{\text{husband}(\text{dilbert}) = \text{false}, \text{single}(\text{dilbert}) = \text{true}\}.$$

What could be the semantics of a non-stratified program?

- Probabilities over well-founded models:
  - Sato, Kameya and Zhou (2005),
  - Hadjichristodolou and Warren (2012).
  - Riguzzi (2015).

- Proposal by Lukasiewicz (2005):
  informally, take the set of every possible probability distributions that satisfy the rules and (probabilistic) facts.

An example

The Dilbert example

$$\mathrm{single}(X) :- \mathrm{man}(X), \textbf{not}\ \mathrm{husband}(X).$$

$$\mathrm{husband}(X) :- \mathrm{man}(X), \textbf{not}\ \mathrm{single}(X).$$

$$0.9 :: \mathrm{man}(\mathrm{dilbert}).$$

- A probability model: $\mathbb{P}(s_1) = 0.1$ and $\mathbb{P}(s_2) = 0.9$ hence $\mathbb{P}(s_3) = 0$.
- Another one: $\mathbb{P}(s_1) = 0.1$ and $\mathbb{P}(s_3) = 0.9$ hence $\mathbb{P}(s_2) = 0$.
- Actually, take any $\gamma \in [0, 1]$:

$$\mathbb{P}(s_1) = 0.1, \quad \mathbb{P}(s_2) = 0.9\gamma, \quad \mathbb{P}(s_3) = 0.9(1 - \gamma).$$

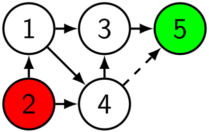
Credal semantics

- Given $\langle \mathbf{P}, \mathbf{PF} \rangle$:
  - A probability model is a probability measure over stable models of the program, such that all probabilistic facts are respected and independent.
  - The set of all probability models is the semantics of the probabilistic logic program.
- Lukasiewicz calls this the "answer-set semantics".
  - More general than answer set programming.
  - Different from answer set semantics (probabilities).
  - We prefer *credal semantics*.
    - Note: another recent semantics based on credal sets by Michels et al. (2015).

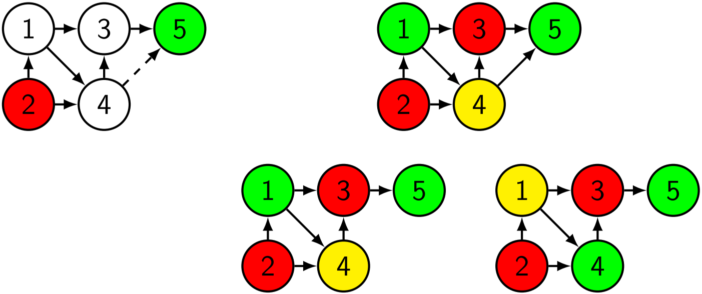
An example: graph coloring

$\text{coloredBy}(V, \text{red}) \text{ :- } \textbf{not } \text{coloredBy}(V, \text{yellow}), \textbf{not } \text{coloredBy}(V, \text{green}), \text{vertex}(V).$
$\text{coloredBy}(V, \text{yellow}) \text{ :- } \textbf{not } \text{coloredBy}(V, \text{red}), \textbf{not } \text{coloredBy}(V, \text{green}), \text{vertex}(V).$
$\text{coloredBy}(V, \text{green}) \text{ :- } \textbf{not } \text{coloredBy}(V, \text{red}), \textbf{not } \text{coloredBy}(V, \text{yellow}), \text{vertex}(V).$
$\text{noClash} \text{ :- } \textbf{not } \text{noClash}, \text{edge}(V, U), \text{coloredBy}(V, C), \text{coloredBy}(U, C).$

vertex(1). vertex(2). vertex(3). vertex(4). vertex(5).
coloredBy(2, red). coloredBy(5, green).
0.5 :: edge(4, 5).
edge(1, 3). edge(1, 4). edge(2, 1). edge(2, 4). edge(3, 5). edge(4, 3).



Inferences



- $\underline{\mathbb{P}}(\text{coloredBy}(1, \text{yellow})) = 0$ and $\overline{\mathbb{P}}(\text{coloredBy}(1, \text{yellow})) = 1/2$.
- $\underline{\mathbb{P}}(\text{coloredBy}(4, \text{yellow})) = 1/2$ and $\overline{\mathbb{P}}(\text{coloredBy}(4, \text{yellow})) = 1$.
- $\underline{\mathbb{P}}(\text{coloredBy}(3, \text{yellow})) = \overline{\mathbb{P}}(\text{coloredBy}(3, \text{yellow})) = 1$.

In the paper: yet another example...

$$\text{wins}(X) \text{ :- } \text{move}(X, Y), \textbf{not } \text{wins}(Y).$$

$$\text{move}(a, b). \quad \text{move}(b, a). \quad \text{move}(b, c). \quad 0.3 :: \text{move}(c, d).$$

$$\begin{tikzcd} a \arrow[r, bend left] & b \arrow[l, bend left] \arrow[r] & c \arrow[r, dashed] & d \end{tikzcd}$$

- move(c, d) is false: unique stable model (where wins(b) is the only winning position);
- otherwise, there are two stable models
  - (wins(c) is true in both of them;
  - wins(a) is true in one,
  - wins(b) is true in the other).
- $\underline{\mathbb{P}}(\text{wins}(a)) = 0.0$ and $\overline{\mathbb{P}}(\text{wins}(a)) = 0.3$;
  $\underline{\mathbb{P}}(\text{wins}(b)) = 0.7$ and $\overline{\mathbb{P}}(\text{wins}(b)) = 1.0$;
  $\underline{\mathbb{P}}(\text{wins}(c)) = 0.3$ and $\overline{\mathbb{P}}(\text{wins}(c)) = 0.3$.

A final example

- If the barber shaves every villager who does not shave himself, does the barber shave himself?

- The program

$$\text{shaves}(X, Y) :- \text{barber}(X), \text{villager}(Y), \textbf{not } \text{shaves}(Y, Y).$$
$$0.5 :: \text{barber}(\text{bob}). \qquad 0.5 :: \text{villager}(\text{bob}).$$

  does not have a stable model.

The structure of credal semantics

Theorem

Given a consistent probabilistic logic program, its credal semantics is a set of probability measures that dominate an infinitely monotone Choquet capacity.

- Infinitely monotone Choquet capacity: belief function, random set...
- Properties:
  - $\underline{\mathbb{P}}(\Omega) = 1 - \underline{\mathbb{P}}(\emptyset) = 1$;
  - for any $A_1, \ldots, A_n$ in the algebra, $\underline{\mathbb{P}}(\cup_i A_i) \geq \sum_{J \subseteq \{1,\ldots,n\}} (-1)^{|J|+1} \underline{\mathbb{P}}(\cap_{j \in J} A_j)$.

Some consequences...

- Credal semantics is a closed and convex set of probability measures.
- We have:

$$\underline{\mathbb{P}}(A) = \sum_{\theta \in \Theta : \Gamma(\theta) \subseteq A} \mathbb{P}(\theta)\,, \qquad \text{(cautious inference)},$$

$$\overline{\mathbb{P}}(A) = \sum_{\theta \in \Theta : \Gamma(\theta) \cap A \neq \emptyset} \mathbb{P}(\theta)\,, \qquad \text{(brave inference)},$$

$$\underline{\mathbb{P}}(A|B) = \frac{\underline{\mathbb{P}}(A \cap B)}{\underline{\mathbb{P}}(A \cap B) + \overline{\mathbb{P}}(A^c \cap B)},$$

$$\overline{\mathbb{P}}(A|B) = \frac{\overline{\mathbb{P}}(A \cap B)}{\overline{\mathbb{P}}(A \cap B) + \underline{\mathbb{P}}(A^c \cap B)}.$$

Inferences

- Inference: computing $\underline{\mathbb{P}}(\mathbf{Q})$ for a set of truth assignments $\mathbf{Q}$.

- An algorithm (Cali et al. (2008)):
  - **Given** a plp $\langle \mathbf{P}, \mathbf{PF} \rangle$ and $\mathbf{Q}$, initialize a and b with 0.
  - For each total choice θ of probabilistic facts, compute the set S of all stable models of $\mathbf{P} \cup \mathbf{PF}^{\downarrow\theta}$, and:
    if $\mathbf{Q}$ is true in every stable model in S, then $a \leftarrow a + \mathbb{P}(C)$;
    if $\mathbf{Q}$ is true in some stable model of S, then $b \leftarrow b + \mathbb{P}(C)$.
  - **Return** a and b (the value of a is $\underline{\mathbb{P}}(\mathbf{Q})$, the value of b is $\overline{\mathbb{P}}(\mathbf{Q})$).

The complexity of inference

Theorem (assume programs are consistent):

Inferential complexity is $\#$NP-equivalent for propositional prob. programs, and $\#$P-equivalent when restricted to stratified propositional prob. programs.

For prob. programs where all predicates have a bound on arity, inferential complexity is $\#\mathsf{NP}^{\mathsf{NP}}$-equivalent, and $\#$NP-equivalent when restricted to stratified programs.

(Counting class $\#$C: class of problems solved by a nondeterministic counting polynomial-time Turing machine with oracle C.)

Query complexity

- Query complexity: the complexity of computing $\mathbb{P}(\mathbf{Q})$ when the input is $\mathbf{Q}$, and the program is fixed.

Theorem (assume programs are consistent):

Query complexity is #P-hard and in #NP; when restricted to stratified programs, it is #P-equivalent.

A summary

	Inferential	Query
Acyclic propositional	#P	—
Acyclic relational (arity-bounded)	#NP	#P
Stratified propositional	#P	—
Stratified relational (arity-bounded)	#NP	#P
Non-stratified propositional	#NP	—
Non-stratified relational (arity-bounded)	$\#\mathsf{NP}^{\mathsf{NP}}$	$\in$ #NP
Non-stratified propositional WELL-FOUNDED	#P	—
Non-stratified relational (arity-bounded) WF	#NP	#P

A word on consistency

Theorem

Consistency checking is in $\mathsf{coNP}^{\mathsf{NP}}$ for propositional and in $\mathsf{coNP}^{\mathsf{NP}^{\mathsf{NP}}}$ for prob. programs where predicates have a bound on arity.

Moving to answer set programming

- Classic negation ($\neg A$), constraints ($:\!-\ \phi$), disjunctive heads:

$$\mathsf{coloredBy}(V, \mathsf{red}) \vee \mathsf{coloredBy}(V, \mathsf{yellow}) \vee \mathsf{coloredBy}(V, \mathsf{green}) :\!-\ \mathsf{vertex}(V).$$
$$:\!-\ \mathsf{edge}(V, U), \mathsf{coloredBy}(V, C), \mathsf{coloredBy}(U, C).$$

Result:

the credal semantics (the set of measures over stable models) of these probabilistic answer set programs is again an infinite monotone credal set.

Conclusion

- Credal semantics is a rather sensible semantics.
- Structure of credal semantics: nicely, infinite monotone Choquet capacities.
- Complexity: partially mapped, still some open questions.